A PROJECT REPORT

on

"BinSense"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

Rohan Jayanti	2005518
Abhas Nayak	20051192
Cyrus Bhandari	20051203
Aayush Singh	20051632

UNDER THE GUIDANCE OF
Mr. Debashis Hati

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

May 2023

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

"BinSense"

submitted by

Rohan Jayanti	2005518
Abhas Nayak	20051192
Cyrus Bhandari	20051203
Aayush Singh	20051632

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2022-2023, under our guidance.

Date: / /

Mr. Debashis Hati
Project Guide

Acknowledgements

We are profoundly grateful to **Mr. Debashis Hati** of **KIIT Deemed to be University** for his expert guidance and continuous encouragement throughout to see that this project reaches its target from its commencement to its completion. .....

Rohan Jayanti

Abhas Nayak

Cyrus Bhandari

Aayush Singh

ABSTRACT

BinSense is a practical and efficient smart dustbin management system developed to enhance waste management in urban areas. The project includes a user-friendly full-stack website featuring interfaces for both municipal authorities and individual users. The user dashboard enables users to subscribe to multiple dustbins and conveniently monitor their current status in real time. Additionally, the admin dashboard empowers administrators to efficiently manage all registered bins and users, including the ability to add or remove bins and oversee user subscriptions. The smart dustbin itself incorporates infrared (IR) sensors to detect nearby individuals and ultrasonic sensors to measure the waste level inside. These bins are seamlessly connected to a live database, ensuring accurate and timely updates of their status. Notably, when a bin reaches its capacity, it intelligently prevents opening upon approach.

By harnessing the potential of embedded systems, BinSense offers a practical solution to streamline waste management practices.

Keywords:

- Smart dustbin
- Embedded systems
- Dustbin management system
- Real-time status
- Full-stack website
- Municipal interface
- Individual user interface
- Subscription management
- Infrared sensors
- Ultrasonic sensors
- Live database
- Waste level detection

Contents

List of Figures

Chapter 1

Introduction

We are living in a world that is in a state of constant upgradation, but there is one ubiquitous problem that we haven't been able to deal with, the problem that is impeding our advancement to a hygienic, clean and healthy society, is garbage. The increase in waste generation has been considered a significant challenge to large urban centers worldwide and it represents a critical issue for countries with accelerated population growth in cities. Mostly in our daily life, we encounter dustbins that are excessively full and garbage spilling out of them. This kind of situation is neither good for our environment nor for our advancement. This problem leads to a huge number of diseases as a large number of insects and mosquitoes breed on the waste accumulated in this garbage. Hence, we developed a project to control the overfilling of the dustbin by making the dustbin smart enough to notify itself for its cleaning. In this project, the smart dustbin management system is built on the microcontroller-based system having ultrasonic sensors on dustbins that will show the current status of garbage on the LCD screens as well as on the mobile app and monitor. The quantity of waste generated and its potential impacts depend on multifarious factors, including the level of industrial development, the way in which wastes are managed, the existing state of the local environment, and the capacity of the receiving media. Nowadays, cities with developing economies experience exhausted waste collection services, and inadequately managed and uncontrolled dumpsites, and the problems are worsening. The key issue of inadequate waste management is that the garbage bin in public places gets overflowed well in advance before the commencement of the next cleaning process.

Hence, we need such a system that can deracinate or at least minimize this problem to some extent. With the advancement in technology, it is high time that we use technology for waste management systems. The Internet of Things and Cloud Computing offer an automation possibility through cyber-physical systems that will change the way waste management is performed. The Smart Dustbin is a singular solution to the specific and peculiar problems of waste management.

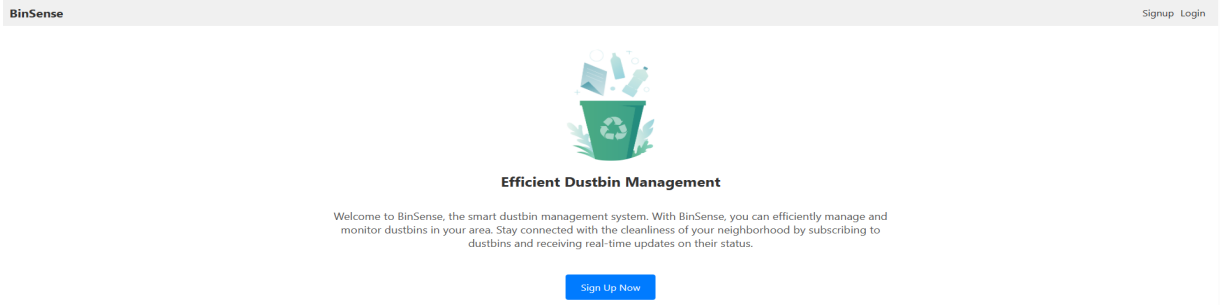


Fig 1.1 Website homepage

Chapter 2

Basic Concepts/ Literature Review

Prior to building the system, the above considerations are taken into account for developing the proposed system. How do the sensors work and how do actuators respond to the input from sensors? IoT is basically machine-to-machine communication. Hence, M2M communications between the machines are discussed.

2.1. PRELIMINARY INVESTIGATION

The first and foremost strategy for the development of a project starts from the thought of designing a platform for the firm which is feasible, portable, accessible, scalable, etc. When this is approved by the organization and our project guide, the first activity i.e.,the preliminary investigation begins. This includes 3 parts:

- Request Clarification
- Feasibility Study
- Request Approval

2.2. REQUEST CLARIFICATION

After the approval of the request to the organization and project supervisor, with an investigation being considered, the project request must be examined to determine precisely what the system requires. In this particular Automated Waste Management Control, we will be using a sensor-based waste management System. R. Suresh, S. Gopinath, K. Govindaraju, T. Devika, N. Suthanthira Vanitha mentioned using an automatic microcontroller-based solid waste management system in which the solid waste is controlled using the GSM module and cloud where the indications of the amount of waste are updated to the municipality corporators on regular basis for efficient management of the garbage. These systems bring a change to the management of field resources as we have used a methodology to overcome dumping, and negligence of stagnant waste that contaminates the area causing dreadful infections and diseases.

2.3. FEASIBILITY STUDY

An important outcome of the preliminary investigation is the determination that the
system request is feasible. This is only possible if it is feasible within the limited resources and time period. The different feasibilities that have to be analyzed are:

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

Operational feasibility:
Operational feasibility deals with the study of prospects of the system to be developed.This system operationally eliminates all the tensions of the admin and helps him in effectively tracking the project's progress. This kind of automation will surely reduce the time and energy previously consumed in manual work. Based on the study,the system is proven to be operationally feasible.

Economic feasibility or the cost-benefit is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning and for lots of purposes, thus the cost of the project hardware is low. Since the system is automation based, the rate of manual interference is comparatively low and leads to lesser consumption or limited consumption of resources without them being wasted.

Since this project uses feasible resources it is much easier to port that into implementation on a one-time investment basis of the required sensors by the government which eventually functions for years with proper maintenance in a systematic manner making a wholesome use of the current technology of waste management.

Technical feasibility:
According to Roger S. Pressman, Technical Feasibility is the assessment of the Technical resources of the organization or a firm or an individual person.The implementation requires a pre-loaded embedded C program according to the desired functioning of the system, and a few of the hardware components that are designed to be connected in a particular manner to acquire desired working and functionality

Economic feasibility:

Economic feasibility or the cost-benefit is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning and for lots of purposes, thus the cost on a project of hardware is low. Since the system is automation based, the rate of manual interference is comparatively low and leads to lesser consumption or limited consumption of resources without them being wasted.Economic feasibility or the cost-benefit is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning and for lots of purposes, thus the cost on a project of hardware is low. Since the system is automation based, the rate of manual interference is comparatively low and leads to lesser consumption or limited consumption of resources without them being wasted.

2.4. REQUEST APPROVAL

Not all the requested projects are desirable or feasible. Some organizations receive so many project requests from client users that only a few of them are pursued. However,those projects that are both feasible and desirable should be put into schedule. After a project request is approved, its cost, priority, completion time, estimation, and personnel requirement are estimated and used to determine where to add it to any project list.Truly speaking, with the approval of those above factors, development works can be
launched.

Chapter 3

Problem Statement / Requirement Specifications

The problem statement of this project is to develop and implement BinSense, a smart dustbin management system that revolutionizes waste management practices in cities. By leveraging embedded systems and modern technologies, BinSense aims to provide municipalities and individual users with a comprehensive solution that enhances waste collection efficiency, reduces environmental impact, and improves overall cleanliness and hygiene in urban areas.

3.1 Project Planning

Features to be developed:

Embedded:

- Automatic opening and closing of the bin's lid when it senses a person in front of it.
- Bin remains closed if it is full, to avoid over-dumping.
- Visual indication for current status
- Sensing the amount of garbage and indicating its levels.
- Connecting to the cloud to store data, monitor and access it via web application by the waste management authorities.

Software:

- User Registration and Authentication:
    - Frontend: User-friendly registration and login interfaces for individual users.
    - Backend: Secure authentication and storage of user credentials.

- User Dashboard:
    - Frontend: Intuitive dashboard displaying subscribed dustbins, their live status, and relevant information.
    - Backend: Integration with the database to fetch and update dustbin status in real time.

- Admin Dashboard:
  - Frontend: Administrative interface to manage dustbins, users, and user's bin subscriptions.
  - Backend: Authorization and authentication for administrators, ability to add/remove bins, view user details, and manage subscriptions.

- Live Dustbin Status Updates:
  - Frontend: Real-time display of dustbin status
  - Backend: Integration with embedded systems in dustbins to collect and update live data in the database.

- Intelligent Dustbin Locking Mechanism:

  - Backend: Logic to prevent dustbin opening when it reaches capacity based on data from embedded sensors.

- Data Security and Privacy:
  Implementing necessary security measures to protect user data and ensure privacy.

- Responsive Design:
  Ensuring the website is accessible and responsive across different devices and screen sizes.

3.2 Project Analysis

In this section, the project requirements and problem statements will be thoroughly analyzed to identify any ambiguities, inconsistencies, or potential mistakes. The goal of this analysis is to ensure a clear understanding of the project scope and to rectify any issues before proceeding with development. The analysis will focus on the following aspects:

3.2.1 Requirement Validation

- Reviewing the collected requirements to ensure they align with the project objectives and address the identified problems.
- Identifying any conflicting or ambiguous requirements that may lead to misunderstandings during development.
- Verifying that the requirements are specific, measurable, achievable, relevant, and time-bound (SMART).

3.2.2 Identifying Assumptions and Constraints

- Identifying any implicit assumptions made during the problem statement formulation or requirement-gathering process.
- Assessing the feasibility and practicality of these assumptions and evaluating their impact on the project.
- Identifying any external constraints, such as time, budget, or technological limitations, that may affect the project implementation.

3.2.3 Gap Analysis

- Comparing the identified requirements with the current state of waste management practices and existing solutions.
- Identifying any gaps or areas where the proposed solution may fall short in addressing the identified problems effectively.
- Analyzing potential risks and challenges associated with the project implementation and assessing their impact on the desired outcomes.

3.2.4 Consistency and Coherence

- Ensuring consistency and coherence across the problem statements, requirements, and proposed solution.
- Checking for any conflicting information or discrepancies between different project components.
- Validating that the proposed solution aligns with the overall project objectives and is in line with best practices and industry standards.

3.2.5 Iterative Refinement

- Acknowledging that the project analysis is an iterative process and may require multiple rounds of refinement.
- Documenting any changes or updates made during the analysis phase and ensuring they are properly communicated and documented.

3.3 System Design

The most creative and challenging phase of the lifecycle is the system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in the implementation of the candidate system. The design may be defined as “the process of applying various techniques and principles for the purpose of defining a device” or it can also be defined as “a process or a system in sufficient details to permit its physical realization”.

The designer's goal is how the output is to be produced and in what format samples of the output and input are also presented. The processing phases are handled through program construction and testing. Finally, details related to the justification of the system and an estimate of the impact of the candidate system on the user and related firm (if any) are documented and evaluated by management as a step toward implementation.

3.3.1 Design Constraints

Constraints, in this context, are limitations or restrictions that impose specific requirements or conditions on the development and implementation processes.
They are classified into two types:

- Software
- Hardware

Software:

1. ARDUINO IDE:
   It is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension '.ino'. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

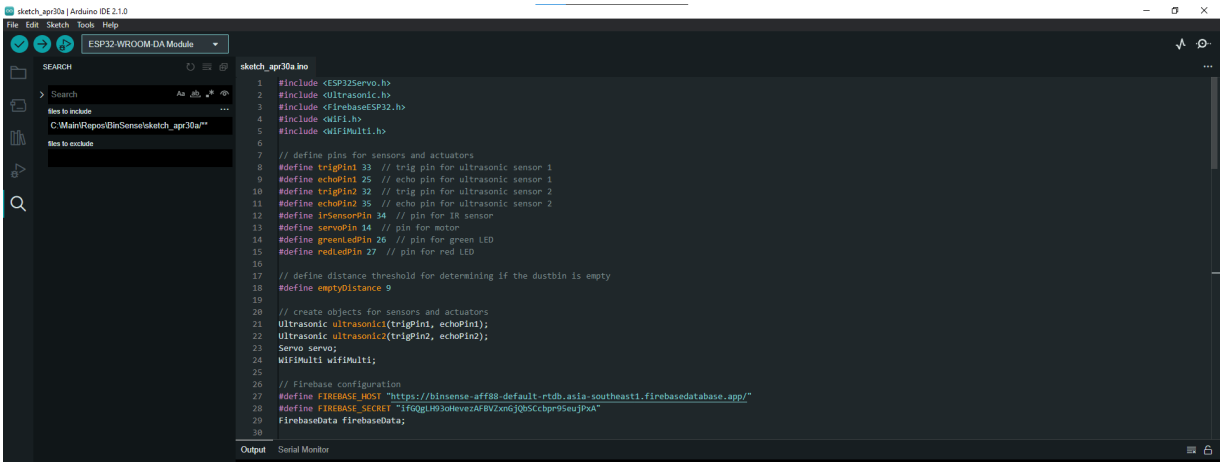


Fig. 3.3.1.a Arduino IDE

2. Visual Studio Code:
   Visual Studio Code (VS Code) is a versatile, open-source code editor developed by Microsoft. It offers a rich ecosystem for software development with extensive language support, extensions for customization, built-in Git integration, and a user-friendly interface. It provides a robust text editor with features like cut/paste, and search/replace, and supports various programming languages. VS Code includes a message area for feedback, and error display, and facilitates saving and exporting. It conveniently displays the configured board and serial port in the bottom right corner. With toolbar buttons, developers can verify and upload programs, create, open, and save files, and access the serial monitor, enabling efficient coding and debugging workflows.

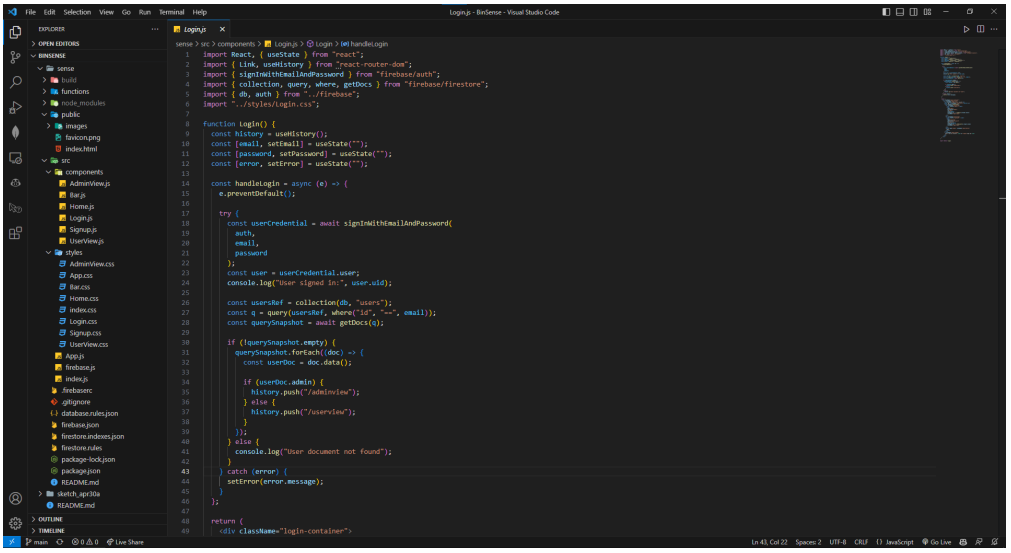


Fig. 3.3.1.b Visual Studio Code

3. Firebase Console:
   Firebase is a versatile open-source platform by Google, designed for developing web and mobile applications. It comprises a comprehensive suite of services and tools that simplify backend development, data management, and hosting.

   Firebase Console serves as the web-based interface for managing and monitoring Firebase projects. It offers a centralized dashboard empowering developers to configure and control various aspects of their Firebase applications.

   Firebase Console provides the following functionalities:
   Project Management
   Authentication
   Realtime Database and Firestore
   Hosting
   Cloud Functions
   Storage
   Analytics and Performance Monitoring

   Firebase Console empowers developers with a centralized hub to effortlessly manage and monitor their Firebase projects. It streamlines the configuration and monitoring of Firebase services, facilitating seamless development, deployment, and maintenance of web and mobile applications.

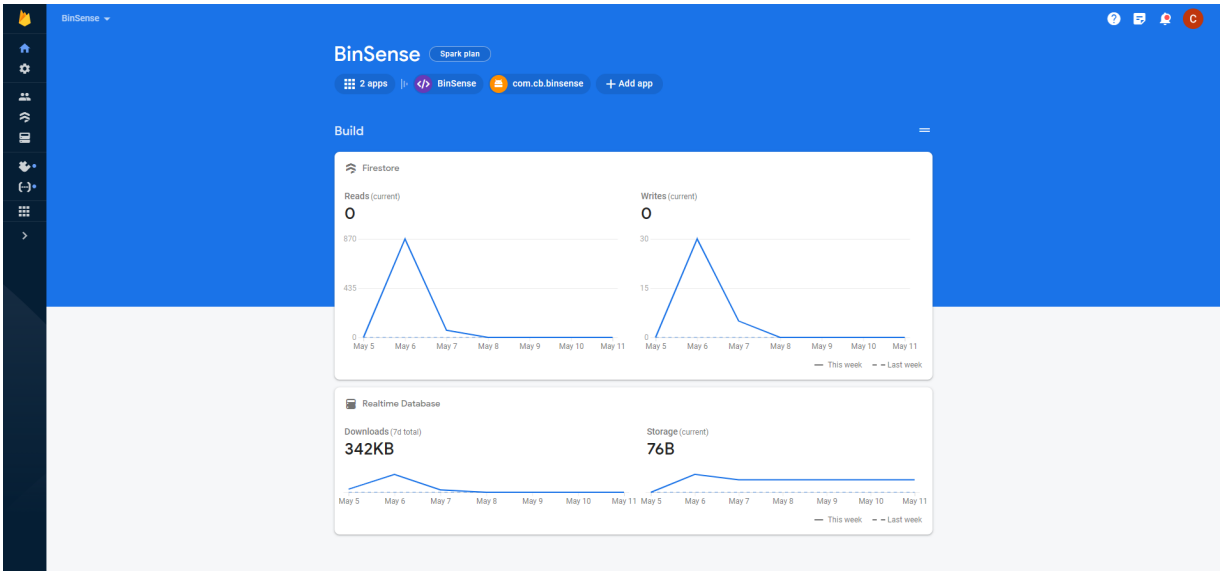


Fig. 3.3.1.c Firebase Console

Hardware:

1. Breadboard:
   A breadboard is a solderless device for temporary prototypes with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connects the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

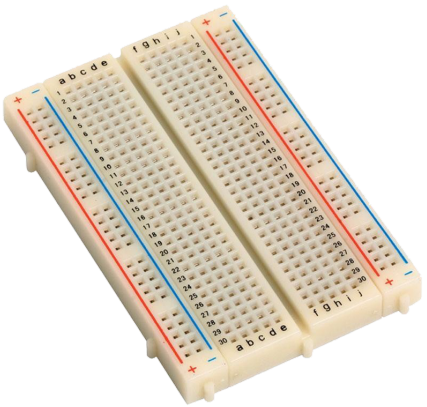


Fig. 3.3.1.d Breadboard

2. ESP32 microcontroller:
   The ESP32 is a powerful microcontroller board based on the ESP32 system-on-chip (SoC) and developed by Espressif Systems. It provides a wide range of functionalities and features for building IoT applications. The board is equipped with numerous digital and analog input/output (I/O) pins that can be connected to various sensors, actuators, and expansion modules.

   With its dual-core processor, the ESP32 offers high processing power and supports Wi-Fi and Bluetooth connectivity. It features a rich set of communication interfaces, including UART, SPI, I2C, and CAN, allowing seamless integration with other devices and systems.

   The ESP32 board provides 38 GPIO pins, which can be configured for various purposes, such as digital input/output, PWM output, or analog input. Additionally, it offers multiple ADC channels, touch sensors, and capacitive touch interfaces, enabling advanced sensing capabilities.

   Programmable with the Arduino IDE, the ESP32 board allows developers to write and upload code easily. It supports both C and C++ programming languages, providing flexibility for application development. The ESP32 can be powered through a USB cable or an external power source, and it operates within a voltage range of 2.2V to 3.6V.

   The hardware design of the ESP32 is open-source, allowing developers to access the layout and production files for customization and modification. It offers extensive documentation, tutorials, and a supportive community, facilitating the development process for IoT projects.
   ESP32 is used instead of Arduino because of the built-in WIFI module, compatibility with Arduino IDE, and overall lower costs.

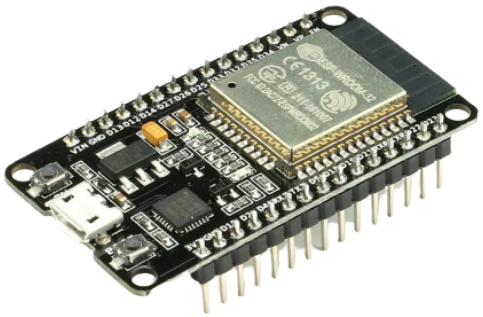


Fig. 3.3.1.e ESP32

3. Servo Motor (MG995):
   A servo motor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analog or digital) representing the position commanded for the output shaft. A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.
   The MG995 motor was used over the popular SG95 because of its high power output and robustness.

Fig. 3.3.1.f MG995 motor

4. Jumper Wires:
   These are connecting wires that allow an electric current to travel from one point on a circuit to another as electric current or electricity needs a medium through which it can move. Often, when building electronics projects, little thought is given to the connecting wire. While it is possible to "get away with" almost anything for many projects, it is sometimes necessary to connect the various electronic components using the right wire. For example, it is often useful to use colored connecting wire to, indicate such items as electronics wire used for connecting the supplies, signals, and grounds. In this way, it is easier to identify the different signals and lines and this reduces the possibility of errors. In addition to this, it is sometimes necessary to have connecting wire of a particular size to ensure the connections are made in the right manner. If the wire is too thick it may not be easy to accommodate in some situations, whereas thicker wire may be needed for higher currents of physical strength or robustness in other situations.

One notable feature of jumper wires is that they often come with plastic caps at the endpoints. These plastic caps can be removed, allowing the wires to be soldered directly onto the circuit board or components. Soldering the wires provides a more secure and permanent connection, especially in projects where stability and reliability are crucial.

Fig. 3.3.1.g Jumper Wires

5. Ultrasonic sensor (HC-SR04):
   The ultra-sonic sensors measure the distance by using ultra-sonic waves. The sensor head emits an ultrasonic wave and receives the waves back from the target. Ultra-sonic sensors measure the distance to the target by measuring the time
   between the emission and reception. Since ultrasonic waves can reflect off a glass or liquid surface and return to the sensor head, even transparent targets can be detected. Detection is not affected by the accumulation of dust or dirt. Presence detection is stable even for targets such as mesh trays or springs. An optical sensor has a transmitter and receiver, whereas an ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables the miniaturization of the sensor head.

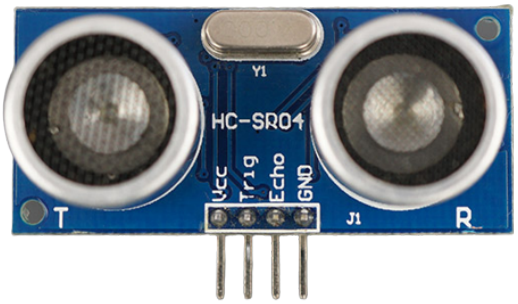


Fig. 3.3.1.h Ultrasonic sensor (HC-SR04)

6. Infrared Sensor (HW-201):
   Infrared (IR) sensors operate by utilizing infrared light to detect the presence or proximity of objects. These sensors emit IR light and then measure the reflection or absorption of the light when it interacts with an object in its path. The distance to the object can be determined by analyzing the intensity of the reflected or absorbed IR light. This measurement is based on the principle of analyzing the strength of the returned signal. The time taken for the IR light to travel from the sensor to the object and back is not typically measured in infrared sensors.

Fig. 3.3.1.i Infrared Sensor (HW-201)

7. LEDs:
   Light-emitting diodes (LEDs) are semiconductor light sources. The light emitted from LEDs varies from visible to infrared and ultraviolet regions. They operate on low voltage and power. LEDs are one of the most common electronic components and are mostly used as indicators in circuits. They are also used for luminance and optoelectronic applications. Based on semiconductor diodes, LEDs emit photons when electrons recombine with holes on forward biasing. The two terminals of LEDs are anode (+) and cathode (-) and can be identified by their size. The longer leg is the positive terminal or anode and the shorter one is the negative terminal. The forward voltage of LED (1.7V-2.2V) is lower than the voltage supplied (5V) to drive it in a circuit. Using an LED as such would burn it because a high current would destroy its p-n gate. Therefore, a current limiting resistor is used in series with the LED. Without this resistor, either low input voltage (equal to forward voltage) or PWM (pulse width modulation) is used to drive the LED. Get details the about internal structure of a LED.

Fig. 3.3.1.j LEDs

3.3.2 System Architecture or Block Diagram

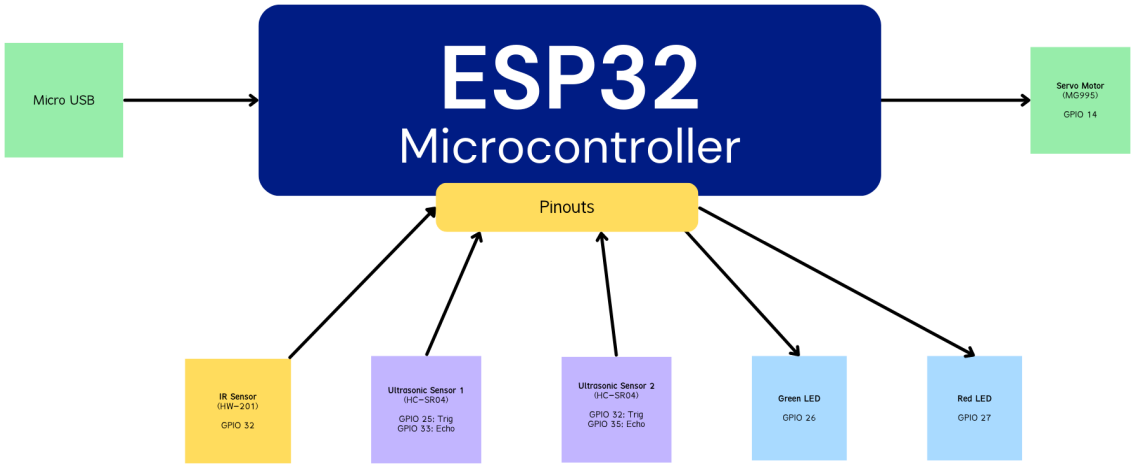


Fig. 3.3.2.a Hardware block diagram

The above block diagram explains the hardware connectivity of sensors and necessary components requiring a power supply from the micro controller ESP32. The arrows represent the input and output representation of the working components. The IR sensor when taken as input detects the motion in front of the bin at agiven height, if it happens to detect any motion then it gives the output to the ESP indicating it to turn on the servomotor.

Similarly, the 2 shown ultra-sonic sensors give the input accordingly as programmed for each component and give the output which activate the either of the mentioned output components, namely, servo motor (for opening and closing the lid of the bin) or LEDs (to indicate the levels of garbage in the bin).

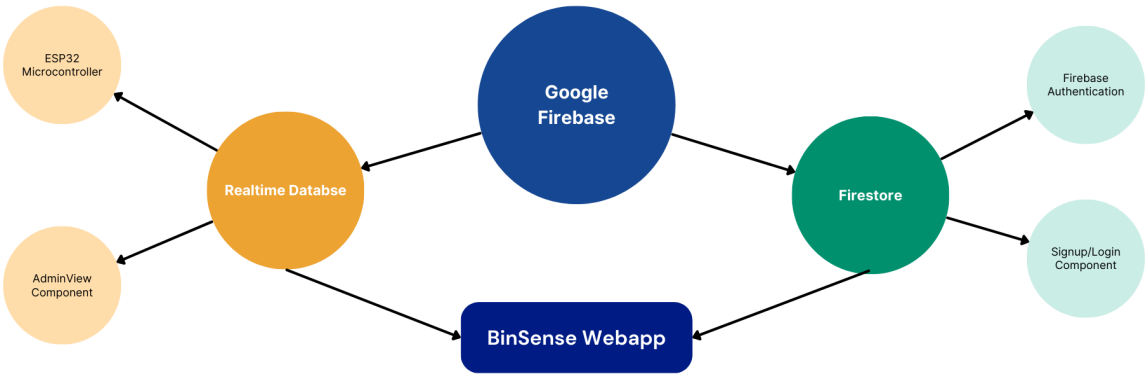


Fig. 3.3.2.b Software block diagram

The above block diagram gives information about the information processing between cloud and application with the data given from sensors through the ESP32's built-in Wi-Fi module. Firstly, the data from the 2 ultra-sonic sensors interfacing with Node MCU are transmitted to the cloud. The data from cloud is then passed to the monitor or the application accordingly with the programmed functioning of the components.

Chapter 4

Implementation

4.1 Methodology OR Proposal

In this project, we describe how to make a Smart Dustbin Management System. The bin itself is made using ESP32 microcontroller, where the lid of the dustbin will automatically open when you approach with trash. The other important components include Ultrasonic Sensor and a MG995 Servo Motor, where the Ultrasonic Sensor is placed inside the dustbin along the side body and when the sensor detects any object, it will trigger the ESP to instruct the motor to open the lid. Similarly, the bin status is conveyed to the app updated realtime through a cloud database. These changes are reflected in the website dashboards.

The website has following components: A home page, Signup Page, Login page, User Dashboard, and an Admin Dashboard.

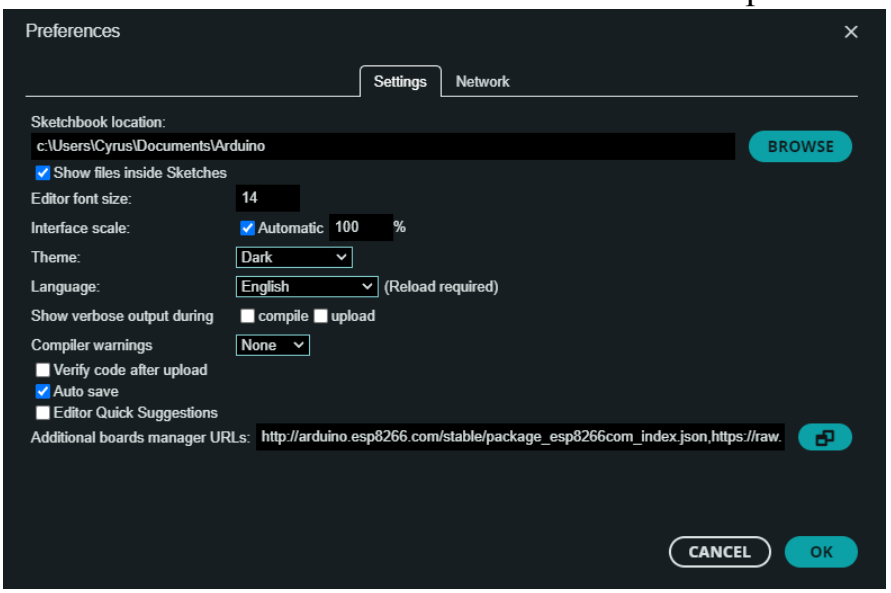
The methodologies and development procedures concerning the hardware and software are as follows:

Software:

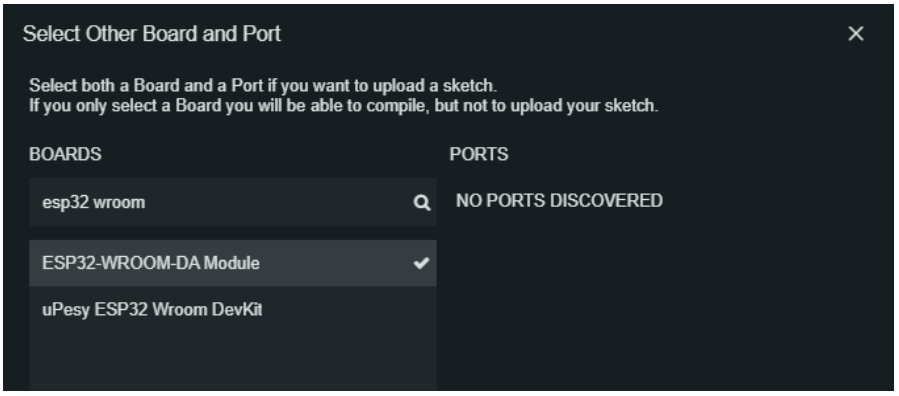
Microcontroller:

- Go to the Arduino IDE Download webpage & download the relevant version for your device. The Arduino IDE software is available for Windows, Mac and Linux operating systems.
- After the setup we are going to get to the initial setup and running of the IDE:
    - Once the installation of the software is done, the Arduino IDE opens into a blank sketch where you can start programming immediately. Firstly, we have to configure the board and port settings to allow us to upload the code to the ESP32 microcontroller.
    - We need the following:
        - CP210x USB to UART Driver
        - Additional board managers: http://arduino.esp8266.com/stable/package_esp8266com_index.json, https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

- Edit the preferences:



- Add the board:



- Install and setup the required libraries from the Library manager:

```
#include <ESP32Servo.h>
#include <Ultrasonic.h>
#include <FirebaseESP32.h>
#include <WiFi.h>
#include <WiFiMulti.h>
```

- Complete the sketch:

```
#include <ESP32Servo.h>
#include <Ultrasonic.h>
#include <FirebaseESP32.h>
#include <WiFi.h>
#include <WiFiMulti.h>

// define pins for sensors and actuators
#define trigPin1 33  // trig pin for ultrasonic sensor 1
#define echoPin1 25  // echo pin for ultrasonic sensor 1
#define trigPin2 32  // trig pin for ultrasonic sensor 2
#define echoPin2 35  // echo pin for ultrasonic sensor 2
#define irSensorPin 34  // pin for IR sensor
#define servoPin 14  // pin for motor
#define greenLedPin 26  // pin for green LED
#define redLedPin 27  // pin for red LED

// define distance threshold for determining if the dustbin is empty
#define emptyDistance 9

// create objects for sensors and actuators
Ultrasonic ultrasonic1(trigPin1, echoPin1);
Ultrasonic ultrasonic2(trigPin2, echoPin2);
Servo servo;
WiFiMulti wifiMulti;

// Firebase configuration
#define                                                    FIREBASE_HOST
"https://binsense-aff88-default-rtdb.asia-southeast1.firebasedatabase.app/"
#define FIREBASE_SECRET "ifGQgLH93oHevezAFBVZxnGjQbSCcbpr95eujPxA"
FirebaseData firebaseData;

void setup() {
 Serial.begin(115200);
 servo.attach(servoPin);
 pinMode(irSensorPin, INPUT);
 pinMode(greenLedPin, OUTPUT);
 pinMode(redLedPin, OUTPUT);

 digitalWrite(greenLedPin, LOW);
 digitalWrite(redLedPin, LOW);
```

```
wifiMulti.addAP("KIIT-WIFI-NET.", "20051203@kiit");
wifiMulti.addAP("Cyrus's iPhone", "enterprise");
wifiMulti.addAP("Xiaomi 12 Pro", "12345678");

// WiFi setup
if (wifiMulti.run() == WL_CONNECTED) {
  Serial.println("Connected to Wi-Fi");
  // Your code here
} else {
  Serial.println("Failed to connect to Wi-Fi");
}
delay(1000);

// Initialize Firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_SECRET);
while (!Firebase.ready()) {
  delay(1000);
  Serial.println("Connecting to Firebase...");
}
```



```
  Serial.println("Firebase connected");
}

void loop() {

  // read the IR sensor value
  int irSensorValue = digitalRead(irSensorPin);

  // if someone is detected by the IR sensor
  if (irSensorValue == LOW) {
    digitalWrite(greenLedPin, HIGH);
    digitalWrite(redLedPin, LOW);
    Serial.println("Person detected");
    // check if the dustbin is empty
    float distance1 = ultrasonic1.read();
    float distance2 = ultrasonic2.read();
    int status = 1;

    Serial.print("Distance 1: ");
    Serial.print(distance1);
    Serial.print(", Distance 2: ");
    Serial.println(distance2);
```

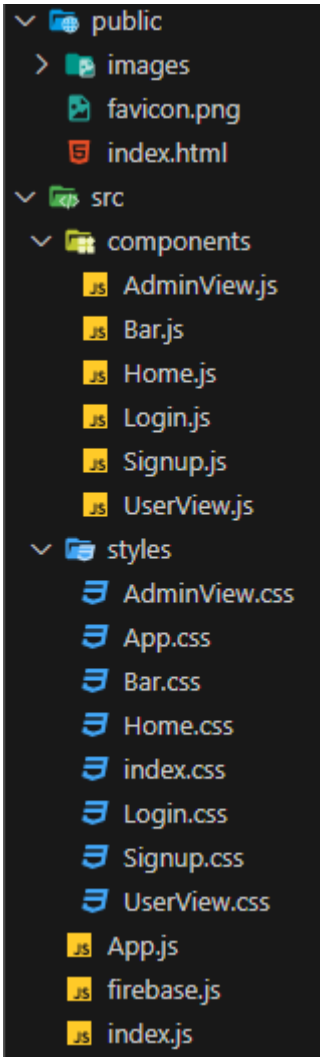
```
  if (distance1 < emptyDistance && distance2 < emptyDistance) {
    Serial.println("Dustbin is not empty");
    digitalWrite(greenLedPin, LOW);
    digitalWrite(redLedPin, HIGH);
    servo.write(0);  // close the bin
    status = 0;
    delay(2000);
  } else {
    Serial.println("Dustbin is empty");
    servo.write(180);  // open the bin
    while (digitalRead(irSensorPin) == LOW) {
      // keep the bin open as long as someone is detected by the IR sensor
      delay(100);
    }
    servo.write(0);  // close the bin
    digitalWrite(greenLedPin, LOW);
    digitalWrite(redLedPin, LOW);
  }

  // update the bin status in Firebase
  Firebase.setInt(firebaseData, "/bins/1/status", status);
  if (firebaseData.dataAvailable()) {
    Serial.println(firebaseData.payload());
  } else {
   BinSense
Serial.println(firebaseData.errorReason());
  }

 } else {
  // if no one is detected by the IR sensor, turn off the LEDs and keep the bin closed
  digitalWrite(greenLedPin, LOW);
  digitalWrite(redLedPin, LOW);
  servo.write(0);
 }
}
```

Website:

- Install NodeJS and Visual Studio Code
- Create a new directory that will store the React App.
- Install the following dependencies to the directory using npm: npx-create-react-app react-router-dom react-sripts react
- The work tree looks like this:



- The approach for building the components:
  The **App.js** component sets up the routes for different pages of the application using react-router-dom. It also handles the authentication state of the user using Firebase's onAuthStateChanged function. If the user is authenticated, they are redirected to the appropriate view (UserView or AdminView), and if not, they are redirected to the login page.

  The **Home.js** component renders the homepage of your application, displaying information about the smart dustbin management system. It includes sections with images, titles, descriptions, and buttons for signing up or logging in.

  The **Signup.js** component provides a form for users to create an account. It uses Firebase's createUserWithEmailAndPassword function to handle user registration.

  The **Login.js** component renders a login form for users to authenticate. It uses Firebase's signInWithEmailAndPassword function to handle user login.

The **AdminView.js** component is intended for the admin user and displays a table of bins and users. It fetches the data from Firebase's Firestore and Realtime Database and allows the admin to delete bins or remove subscriptions for users.

The **UserView.js** component is for regular users and displays a table of bins and their statuses. It fetches the data from Firebase's Realtime Database and allows the user to remove bins. In order to add a bin a unique binID and secret have to passed. Firebase validates this and returns the required data.

The components are ensured to work in cohesion with each other. The states and props are properly transferred between them.

The Firebase utilities used in the app provides server-side and database logic to the application. We will look at Firabase in the next section.

- The UI was also made with similar consideration of each component individually as well as the system as a whole for cohesion.

Backend:

- The development process for the backend of our website using Firebase, Firestore, Realtime Database, and Authentication involves the following steps:

- Firebase project setup: We start by creating a new Firebase project and configuring it according to our requirements. This involves creating a project in the Firebase console, enabling the necessary services, and obtaining the configuration details required to connect our backend to Firebase.

- Firestore setup: We set up Firestore, which serves as our NoSQL document database. We define the structure of our collections and documents to store user data. For example, we might have a "users" collection where each document represents a user and contains fields like admin, id, and subscriptions.

- Realtime Database setup: We configure the Realtime Database to handle real-time updates of the dustbin state. We define the necessary database structure to store and update the bin states as they change. The Realtime Database is updated via an ESP32 microcontroller, which sends updates to Firebase whenever there is a change in the dustbin state.

- Authentication setup: We enable the Firebase Authentication service and choose the email-password sign-in method for user authentication. This allows users to register and log in to our website using their email and password credentials. Firebase Authentication takes care of securely storing and managing user credentials, session management, and user authentication workflows.

- User management with Firestore: We utilize Firestore to store user data such as the user's admin status, unique ID, and bin subscriptions. The subscriptions field holds the IDs of the bins that the current user is subscribed to and can monitor. This allows us to associate user-specific data with their Firestore document and manage user subscriptions effectively.

- ESP32 integration with Realtime Database: We set up the ESP32 microcontroller to send updates to the Realtime Database whenever there is a change in the dustbin state. This involves integrating the Firebase Realtime Database SDK into the ESP32 firmware and establishing a secure connection with Firebase. The ESP32 updates the relevant bin state in the Realtime Database, and these changes are reflected on the website wherever the relevant bins are displayed.

- Security rules and data validation: We configure security rules for Firestore and Realtime Database to define access permissions and data validation rules. These rules ensure that only authenticated users can access and modify their own data and enforce data integrity and validation constraints. We define rules to prevent unauthorized access to sensitive information and maintain data consistency.

- Error handling and logging: We implement error handling mechanisms to gracefully handle exceptions and unexpected scenarios. We log relevant information to assist with debugging and troubleshooting. Firebase provides logging and error reporting features that help capture and analyze errors and exceptions occurring in the backend.

- Throughout the development process, it's important for us to document our backend code, configurations, and any important decisions made. This documentation will aid in future maintenance, collaboration, and troubleshooting.

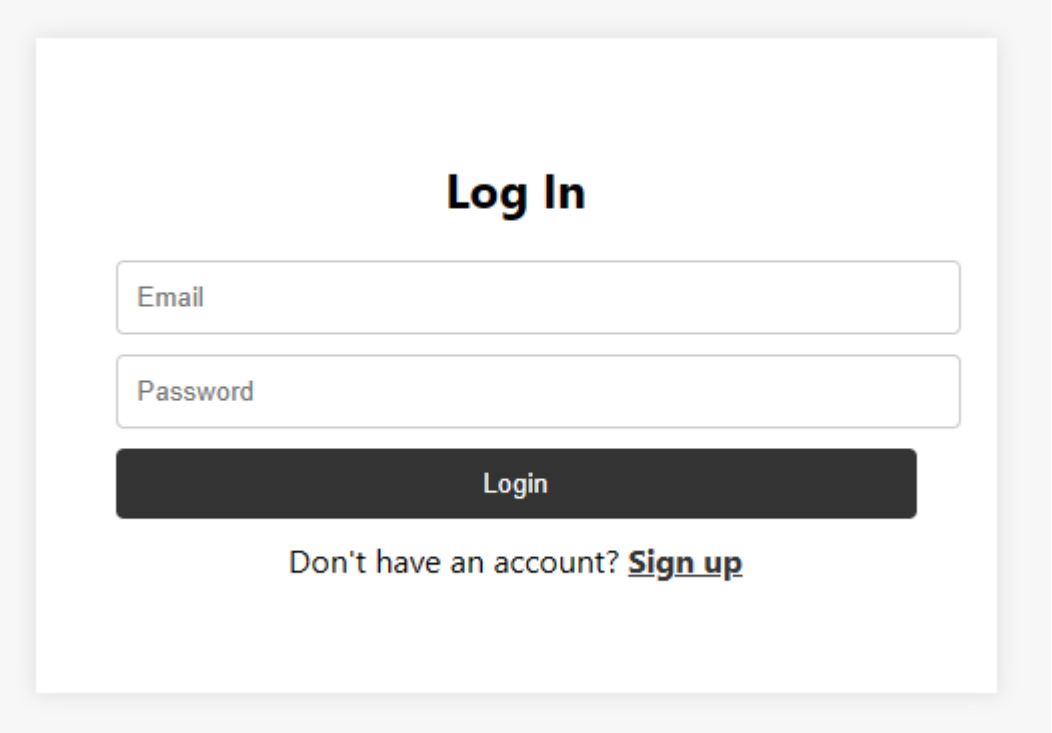
4.2 Testing OR Verification Plan

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Admin Access	An admin logs in to the website. If the credentials are accurate, the admin should be directly routed to the admin dashboard.	/adminview route opens	/adminview route opens
T02	User Access	A user logs in to the website. If the credentials are accurate, the admin should be directly routed to the user dashboard.	/userview route opens	/userview route opens
T03	User Signup	User data added to Firestore	Firestore updates and user can login	Firestore updates and user can login

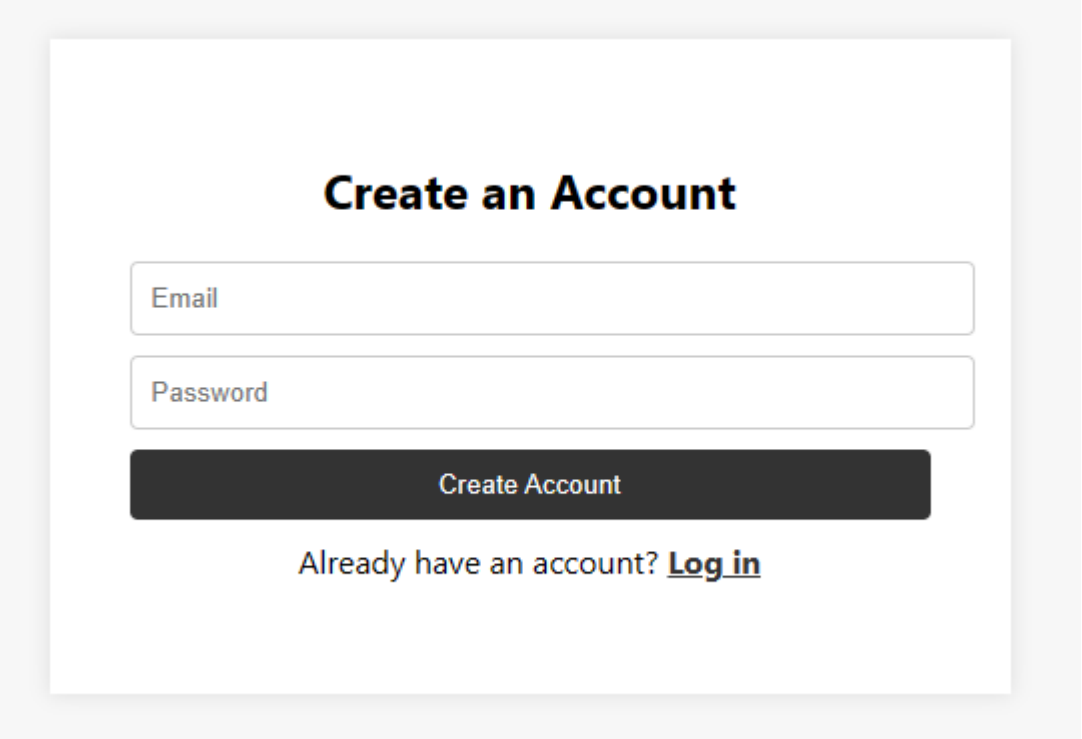
Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T04	Bin Status Update	When the ultrasonic sensors detect that the bin is filled, the ESP32 communicates this with the Realtime database.	Bin status updates to ‘0’ from ‘1’. The dashboard updates from ‘1’to ‘Offline’.	Bin status updates to ‘0’ from ‘1’. The dashboard updates from ‘1’to ‘Offline’.
T05	Admin actions: Deleting bins and user subscriptions	From the admin dashboard, one of the added bins is removed. The subscriptions of a user are also removed.	The table entry of the bin is removed, the data is removed from the Realtime database, the user subscriptions are updated to be empty in Firestore.	The table entry of the bin is removed, the data is removed from the Realtime database, the user subscriptions are updated to be empty in Firestore.
T06	User actions: Bin added with credentials and then removed	A user adds ‘Bin 1’ by passing the binid and secret. The bin is added to the table view. Firestore updates the user subscriptions. User then removes the bin.	Firestore and user dashboard changes are seen immediately.	Firestore and user dashboard changes are seen immediately.
T07	User attempts to add bin with invalid credentials	A user logs in and attempts to add a bin using an incorrect secret.	Error logged and shows alert “Invalid bin ID or secret”	Error logged and shows alert “Invalid bin ID or secret”

4.3 Result Analysis OR Screenshots

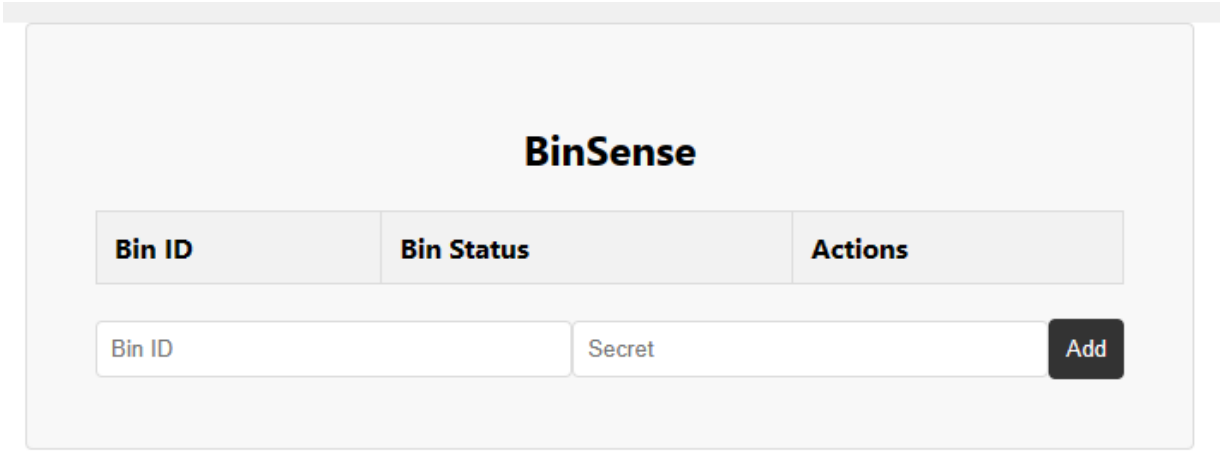
Login page:



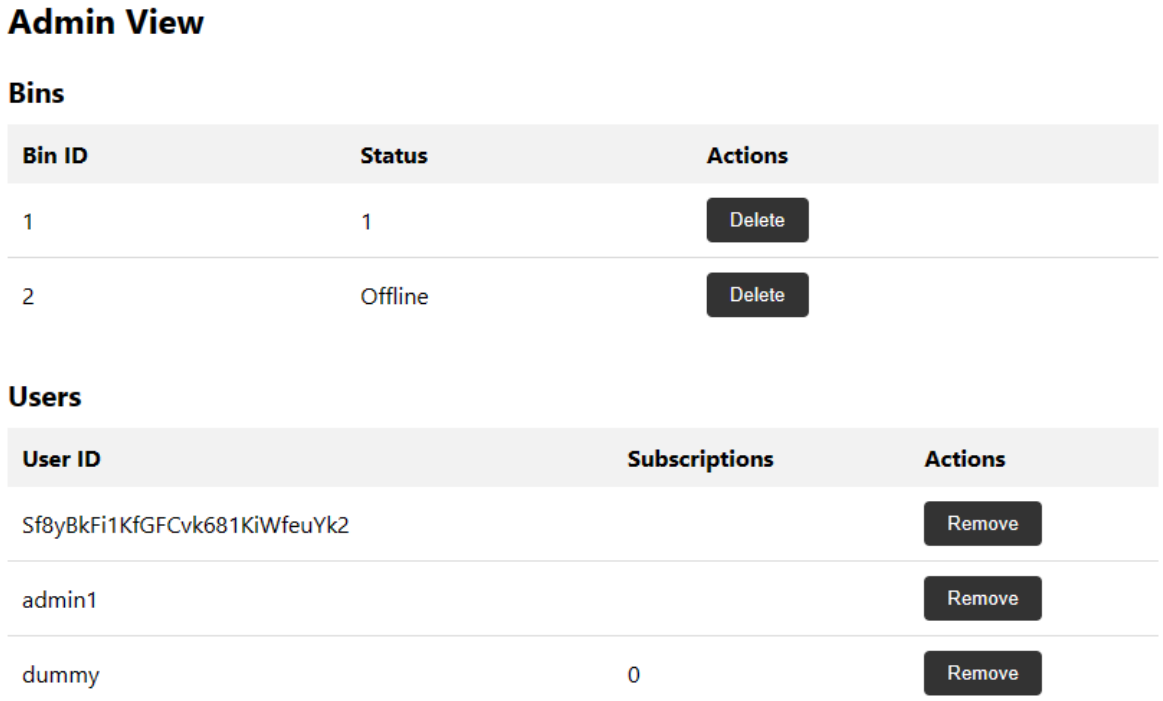
Signup page:



Userview:



Adminview:



4.4 Quality Assurance

The quality of the project is assured by our respected minor project mentor Mr. Debashis Hati.

Chapter 5

Standards Adopted

5.1 Design Standards

In our project, we emphasize the importance of following sound design practices to ensure a well-structured and maintainable system. Wee prioritize the following recommended practices:

- Employing clear and consistent naming conventions for variables, functions, and classes to enhance code readability and maintainability.

- Breaking down complex tasks into smaller, manageable functions or modules to promote code reusability and ease of maintenance.

- Utilizing appropriate code documentation to facilitate understanding and collaboration among team members.
- Considering principles of software design, such as modularity, encapsulation, and separation of concerns, to enhance system flexibility and scalability.

5.2 Coding Standards

To maintain code quality and readability, we follow a set of coding standards and best practices. We emphasize the following guidelines:

- Writing concise and readable code by using appropriate indentation and formatting.

- Using meaningful and descriptive names for variables, functions, and classes.

- Keeping functions and methods focused on a single task or responsibility for improved code maintainability.

- Encouraging the use of comments to provide additional context and explanations when necessary.

- Applying consistent coding styles and conventions across the project to enhance code readability and collaboration.

5.3 Testing Standards

We recognize the importance of testing and quality assurance in our project. We prioritize the following practices:

- Conducting thorough testing at various levels, including unit testing, integration testing, and system testing, to ensure proper functionality and detect defects.

- Creating well-designed test cases that cover different scenarios and edge cases to maximize test coverage.

- Utilizing appropriate testing tools and frameworks to automate testing processes and increase efficiency.

- Regularly reviewing and refining our testing approach based on feedback and lessons learned from previous testing cycles.

- Documenting and reporting any discovered defects or issues to facilitate their resolution.

Chapter 6

Conclusion and Future Scope

6.1 CONCLUSION

The automatic waste management system using ESP32 has been experimentally proven to work satisfactorily. The webapp for the BinSense system is also demonstrated to work correctly. We could see the live status updates, functional databse queries, and autrhentication.

Smart dustbins are the now the needs of Smart buildings. Smart waste monitoring and
management is the keen idea of smart city planners. This implementation of smart
garbage Bin indicator receptacle, gives a solution for unsanitary environmental condition in a city. This implementation of Smart Garbage collection bin using internet, Ultrasonic sensor, Servo motor and ESP32. This system assures to send status on dashboard of dustbins when the garbage level reaches its maximum. The record can be sent to the higher authoritybwho can take appropriate action. Therefore, the smart garbage management system makes the garbage collection more efficient. The use of solar panels in such systems may reduce the energy consumption.

6.2 FUTURE SCOPE

Using this system as framework, the system can be expanded to include various other options which could include mobile application control of motor and wi-fi controlledmonitoring. These will expand the working capability and efficiency of this prototype.
The main aim of this project is to reduce human resources and efforts along with the enhancement of a smart city vision. We have often seen garbage spilling over from dustbins on to streets and this was an issue that required immediate attention. The proverb “Cleanliness is next to god and clean city is next to heaven” inspired us to conceptualized the project. Smart dustbin helps us to reduce the pollution. Many times, garbage dustbin is overflow and many animals like dog or rat enters inside or near the dustbin. This project can avoid such situations. And the message can be sent directly to the cleaning vehicle instead of the contractor’s office. Swatch Bharat Abhiyan (English: Clean India Mission and abbreviated as SBA or SBM for "Swatch
Bharat Mission") is a national campaign by the Government of India, covering 4,041 statutory cities and towns, to clean the streets, roads and infrastructure of the country.

In our system, the Smart dustbins are connected to the internet to get the real time information of the smart dustbins. In the recent years, there was a rapid growth in population which leads to more waste disposal. So, a proper waste management system is necessary.

References

[1] Ruhin Mary Saji, Drishya Gopakumar, Harish Kumar, K N Mohammed sayed, Lakshmi S (2016) “A Survey on Garbage Management in cities using IOT”, International Journal of Engineering and Computer Science, Vol.5, Issue.11, ISSN:2319-7242, pp.18749-18754.
[2] Shilan Abdulla Hassan, Noor Ghazi M.Jameel, Boran sekeroglu (2016), “Smart solid waste monitoring and collection system”, International Journal of Advanced Research in computer science and software engineering, Vol.6, Issue.10, ISSN: 2277128X, pp.7-12.
[3] Shyamala S C, Kunjan Sindhe, Viswanth Muddy, Chitra C N (2016), “Smart waste management system”, International Journal of Scientific Development and Research, Vol.1, Issue.9, ISSN: 2455-2631, pp.224-230.
[4] Kannapiran S, Arvind Chakrapani, "A Novel Home Automation System using Bluetooth and Arduino", International Journal of Advances in Computer and Electronics Engineering, Volume 2, Issue 2, February 2017, pp. 41 - 44.
[5] Kasliwal Manasi H, Suryawanshi Smithkumar B (2016) “A Novel approach to Garbage Management Using Internet of Things for smart cities”, International Journal of Current Trends in Engineering & Research, Vol.2, Issue.5, pp.348-353, e-ISSN 2455-1392.
[6] Meghana K C, Dr. K R Nataraj (2016) “IOT Based Intelligent Bin for smart cities”, International Journal on Recent & Innovation Trends in Computing and Communication, Vol. 4, Issue.5, ISSN:2321-8169.
[7] Kanchan Mahajan, Prof.J.S.Chitode (2014), “Waste Bin monitoring system using Integrated Technologies” International Journal of Innovative Research in Science, Engineering and Technology, Vol.3, Issue.7, ISSN: 2319-8753.
[8] Narayan Sharma, Nirman singha, Tanmoy Dutta (2015), “Smart bin implementation for smart cities”, International Journal of Scientific and Engineering Research, Vol.6, Issue.9, ISSN: 2229-5518, pp.787-791.
[9] Monika K A, Nikitha Rao, Prapulla S B, Shobha G (2016) “Smart dustbin – An Efficient Garbage Monitoring System”, International Journal of Engineering Science and Computing, Vol.6, Issue.6, DOI:4010/2016.1694, ISSN:2321 3361, pp.7113-7116.

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

BinSense

Rohan Jayanti
2005512

Abstract: The aim of our project is to develop an Automated Waste Management System that utilizes IoT technology to efficiently manage solid waste. The objective is to provide a system that can monitor and manage waste in real-time, reducing manual labor and promoting environmental sustainability.

Individual contribution and findings: Hardware Setup and Integration Introduction
BinSense is a smart waste management system that aims to automate the process of waste collection and disposal. It utilizes embedded systems and sensors to enable dustbins to open and close their lids automatically when they detect a person in front of them. The system also integrates with the cloud, allowing waste management authorities to monitor and analyze the data collected by the sensors in real-time.

Individual Contribution:

As a member of the team responsible for hardware setup and integration, my main contribution to the project was designing and implementing the embedded systems that enable the dustbins to operate automatically. This involved selecting and integrating the necessary components such as sensors, microcontrollers, and motors. I was also responsible for integrating the dustbins' sensors with the cloud, ensuring that data could be stored and monitored via a web application.

Challenges Faced:

During the hardware setup and integration phase, several challenges were encountered. One of the primary challenges was the voltage requirements of the components, which were different from what the microcontroller could provide. This necessitated the use of an external battery setup. To link the components to the battery, a breadboard was used as a common circuit. Initially, a voltage regulator was used, but it failed to support the system. This required us to re-evaluate the design and find alternative solutions.
Another significant challenge we encountered was with the motor that controlled the opening and closing of the dustbin lids. Despite receiving power and signal, the motor was not working correctly. After several rounds of troubleshooting, we discovered that the negative terminals of the motor, microcontroller, and battery setup had to be synced together on the breadboard. This allowed the motor to function correctly and the dustbin lids to operate automatically.

Solutions Developed:

To overcome the challenges faced during the hardware setup and integration phase, we developed several solutions. For the voltage requirements challenge, we decided to use an external battery setup that could provide the necessary power to the components.

The voltage regulator was salso replaced with a more reliable and efficient one. Additionally, I redesigned the circuit to optimize power consumption and prevent overloading. For the motor issue, we conducted extensive testing to identify the root cause of the problem. After syncing the negative terminals of the motor, microcontroller, and battery setup on the breadboard, we were able to get the motor to work correctly. We also implemented several safety features to prevent damage to the motor or other components. Conclusion In conclusion, I am proud of the contribution I made to the hardware setup and integration of BinSense. Despite encountering several challenges, we were able to develop solutions that enabled the dustbins to operate automatically and transmit data to the cloud securely. Through collaboration with other team members, we were able to create a smart waste management system that has the potential to revolutionize the way waste is collected and managed in urban areas.

Individual contribution to project report preparation: I prepared Chapter 1 and 2 of the project report.

Individual contribution for project presentation and demonstration: I prepared the "Features" and "Benefits" portion of the presentation.

Full Signature of Supervisor: Full signature of the student:

……. ……..

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

BinSense

Abhas Nayak
20051192

Abstract: The aim of our project is to develop an Automated Waste Management System that utilizes IoT technology to efficiently manage solid waste. The objective is to provide a system that can monitor and manage waste in real-time, reducing manual labor and promoting environmental sustainability.

Individual contribution and findings: Arduino Setup and Integration

As a member of the team responsible for the Arduino setup and integration, my main contribution to the project was ensuring the seamless integration of the Arduino with the other components of the system. This involved setting up the Arduino microcontroller board, configuring the sensors, and developing the necessary code to interact with the cloud-based database.

One of the major challenges we faced during the Arduino setup was configuring the sensors to detect the presence of waste accurately. We spent a considerable amount of time calibrating the sensors and ensuring that they were sensitive enough to detect the presence of even small amounts of waste. We also had to ensure that the sensor readings were reliable and consistent to avoid false alarms or missed detections.

Once we were satisfied with the sensor readings, we proceeded to integrate the Arduino with the other components of the system. To achieve this, we had to develop the necessary code to enable the Arduino to communicate with the cloud-based database. This involved setting up the Wi-Fi connection and configuring the Firebase SDK.

One of the challenges we faced during the setup of the Arduino was setting up the Wi-Fi connection. We had to ensure that the Arduino was able to connect to the Wi-Fi network securely and reliably, as any disruptions in the connection could lead to missed sensor readings or delayed data transmission. We also had to ensure that the Wi-Fi credentials were secure and protected against potential attacks.

After setting up the Wi-Fi connection, we proceeded to configure the Firebase SDK. This involved creating a Firebase account and setting up a Realtime Database to store the sensor readings. We also had to configure the Firebase Authentication feature to ensure that only authorized users could access the system. Once the SDK was set up, we had to develop the necessary code to interact with the database and ensure that the data from the sensors was being accurately sent to the database in real-time.

One of the major challenges we faced during the integration process was ensuring that the Arduino and the website were communicating with each other seamlessly. We had to ensure that the website was able to fetch and display the real-time data sent by the Arduino accurately. We also had to make sure that the website was able to send commands to the Arduino to trigger actions such as opening and closing the dustbin lid.

To achieve this, we had to develop a communication protocol that allowed the website and the Arduino to exchange data and commands in real-time. We used the Firebase Realtime Database as the intermediary for this communication, with the website reading from and writing to the database and the Arduino reading from and writing to the database as well.

One of the major advantages of using the Firebase Realtime Database was its ability to handle real-time data updates. This meant that the website could display the sensor readings and the dustbin status in real-time, providing users with up-to-date information on the status of the system.

Overall, my experience working on the Arduino setup and integration was extremely rewarding. While there were some challenges along the way, I was able to apply my knowledge of embedded systems and programming to develop a solution that integrated seamlessly with the other components of the system. By working collaboratively with the other members of the team, we were able to overcome the challenges we faced and deliver a product that met the requirements of the project.

It is important to note that while the data collected by the Arduino was sent via the Firebase SDK, it was not sent to cloud servers for any kind of analytics. Our focus was on developing a system that was efficient in collecting data and communicating with other components of the system in real-time. With the successful integration of the Arduino into the BinSense system, we were able to create a reliable and efficient system for waste management.

Individual contribution to project report preparation: I prepared Chapter 3 of the project report.

Individual contribution for project presentation and demonstration: I prepared a slide containing the “Introduction” part and also assisted Cyrus Bhandari in preparing the “Working” part of the presentation.

Full Signature of Supervisor: Full signature of the student:

……. ……..

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

BinSense

Cyrus Bhandari
20051203

Abstract: The aim of our project is to develop an Automated Waste Management System that utilizes IoT technology to efficiently manage solid waste. The objective is to provide a system that can monitor and manage waste in real-time, reducing manual labor and promoting environmental sustainability.

Individual contribution and findings: Frontend Development

As a member of the frontend development team for the BinSense project, my primary responsibility was to design and develop the user interface of the system. Our team was tasked with creating a website that would allow users to interact with the system and access its various features. To achieve this, we developed several components, including the homepage, signup page, login page, navbar, user dashboard, and admin dashboard. Each of these components had a specific purpose and was designed to be intuitive and easy to use for both individual users and administrators. The homepage was designed to provide a brief overview of the BinSense system and its main features. It featured a clean and modern design with a hero image, prominent call-to-action buttons, and a concise explanation of how the system works. The signup and login pages were created to allow users to create an account or log in to an existing account. We made sure that these pages were user-friendly and provided clear instructions for users to follow. The navbar was designed to provide easy navigation throughout the website. It featured links to the homepage, user dashboard, and admin dashboard. Additionally, we included a search bar that allowed users to quickly search for specific dustbins or locations. The user dashboard was designed to provide individual users with a summary of their account activity and information about the dustbins they were managing. Users could view their account details, track their waste generation, and view their past transactions. We also included a feature that allowed users to request a pickup of their waste. The admin dashboard was designed to provide administrators with an overview of the system's activity. Admins could view all the dustbins on the map, manage user accounts, view transaction history, and generate reports. This dashboard was designed to be comprehensive and provide admins with all the information they needed to effectively manage the system. One of the biggest challenges we faced during the development process was ensuring that the website was responsive and accessible across different devices and screen sizes. To achieve this, we implemented a responsive design approach that allowed the website to adapt to different screen sizes and orientations. We made sure that the website was optimized for desktop, tablet, and mobile devices. Another challenge was ensuring that the website was secure and protected against potential attacks. To achieve this, we integrated Firebase Authentication into the website, which allowed us to securely manage user authentication and access control. We also implemented several security features, such as form validation and input sanitization, to prevent potential attacks.

We worked closely with the backend development team to ensure that the frontend was properly integrated with the backend. This involved testing the website with various use cases and ensuring that it was able to effectively communicate with the backend API. We made sure that the website was performing optimally and that there were no issues with the communication between the frontend and backend. Overall, I am proud of the work our team has done on the frontend of the BinSense system. We have created a user-friendly and intuitive website that effectively communicates with the backend API and provides users with the features they need to effectively manage their waste. Our website is responsive, secure, and accessible across different devices and screen sizes. I am confident that our team has created a website that will have a positive impact on the environment and help reduce waste.

Individual contribution to project report preparation: I prepared Chapter 4 of the project report.

Individual contribution for project presentation and demonstration: In collaboration with Abhas Nayak I prepared the ‘Working” part of the presentation..

Full Signature of Supervisor:
…….

Full signature of the student:
…….

SAMPLE INDIVIDUAL CONTRIBUTION REPORT:

BinSense

Aayush Singh
20051632

Abstract: The aim of our project is to develop an Automated Waste Management System that utilizes IoT technology to efficiently manage solid waste. The objective is to provide a system that can monitor and manage waste in real-time, reducing manual labor and promoting environmental sustainability.

Individual contribution and findings: Backend Development and Firebase Integration

As a backend developer on the BinSense project, I was responsible for designing and developing the server-side of the application. The backend of the system was built using Firebase, which is a platform that provides a suite of services for building web and mobile applications. Firebase provides a complete backend solution, including authentication, cloud storage, and real-time databases, which made it an ideal choice for our project.

One of my first tasks was to configure Firebase for our project. This involved setting up the Firebase account, creating a new project, and configuring the Realtime database and Firestore. Time was spent configuring both the Realtime database as well as Firestore, which are used for storing user data and dustbin information. We chose to use Firestore for storing user data because it offers more advanced querying and scalability features.

Since the scale of the system wasn't massive, I ensured that the implementation offered high scalability by following best practices and keeping the design modular. To accomplish this, I created a serverless architecture, which means that the backend services do not run on a server, but rather are triggered by an event. This architecture is highly scalable and cost-efficient since it only incurs charges when an event occurs.

Additionally, I implemented the logic for authentication, role-based classification, document, and table queries. Firebase provides easy-to-use APIs for all these functionalities, which made the development process smoother. While APIs were not exactly used in the system, I added and configured the Firebase SDK to the React app (website), allowing the website to communicate with the backend services seamlessly.

The primary purpose of the backend development was to handle the authentication, role-based classification, and document and table queries. As such, the backend was developed using Firebase services. The Firebase SDK was added and configured to the React app (website), allowing for seamless communication between the website and the backend services.

Although the scale of the system wasn't massive, the implementation offered high scalability by following best practices and keeping the design modular. The use of Firebase's services allowed for the efficient and straightforward development of the backend, which resulted in a robust and scalable system.

In conclusion, my contribution to the BinSense project's backend development and Firebase integration was crucial in enabling the system to function as intended. The use of Firebase's services allowed for the efficient and straightforward development of the backend, which resulted in a robust and scalable system.

The implementation of a serverless architecture further ensured that the system was highly scalable and cost-efficient. The configuration of both the Realtime database and Firestore allowed for the storage of user data and dustbin information, respectively. The addition and configuration of the Firebase SDK to the React app allowed for seamless communication between the website and the backend services.

Individual contribution to project report preparation: I prepared Chapter 5 and 6 of the project report.

Individual contribution for project presentation and demonstration: I prepared the "Future Scope" portion of the presentation.

Full Signature of Supervisor:
…….

Full signature of the student:
…….

TURNITIN PLAGIARISM REPORT

ORIGINALITY REPORT

20%	16%	10%	19%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.kscst.iisc.ernet.in Internet Source	11%
2	Submitted to National Institute of Technology Warangal Student Paper	3%
3	Submitted to University of Derby Student Paper	2%
4	ir.aiktclibrary.org:8080 Internet Source	1%
5	mobilityresearchforum.com Internet Source	1%
6	forum.arduino.cc Internet Source	1%
7	gitlab.eclipse.org Internet Source	<1%
8	M. Srilatha, C. Abhinav, M. Balaram, A. Sanjana. "Smart Monitoring and Collection of Garbage System Using Internet of Things", 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021 Publication	<1%
9	gujaratresearchsociety.in Internet Source	<1%
10	Submitted to Ho Chi Minh University of Technology and Education Student Paper	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		